3.7周一

C#中，string、类、数组 都是引用类型，string s = “hello”；string s =“你好” 在堆中开辟两块空间分别存放hello和你好，在栈中指向这个s，hello不会被释放，占据内存。

string —>char 用tocharArray方法。StringBuilder适用于字符密集操作型，只会在开辟的内存空间中执行操作。（c# P100-P104）

本周任务：仔细看opc程序，包括导包，内置函数和如何与L1连接。

3.8周二

上午：学习继承，子类：派生类 父类：基类。子类能继承父类属性和方法（方法名重名时用子类的），但继承不了私有字段，构造函数是先执行父类，再执行自己，默认调无参的。

抽象类、虚方法和接口：其实都是为了实现多态

虚方法：virtual，方法体可以为空。

抽象类：abstract ; 只有方法名，不能有方法体，且必须在子类中重写，不能实例化，抽象类的作用就是给子类继承。

接口：接口只能包含方法，举例：鸟会飞，飞机也会飞——接口表示一种能力。接口用于能力相同，但其余性质完全不同的东西。子类可以继承父类和接口（先父类后接口），接口也不能实例化（和抽象类一样）

下午：opc程序，opc是中间件（上位机和下位机连接的桥梁），opcautomation需引用OPCDAAuto.dll opcserver->opcgroups->opcgroup->opcitems->opcitem。opcserver.connect(服务器名称[string]，对象节点)

opc调用分为异步、同步和订阅。订阅：opcgroup.DataChange += 事件（事务id，返回数据数，句柄，数据值，通信质量，时间戳）。订阅是第一次读所有变量，后面只有当value值变化时才会触发事件。\*Array索引遍历数据时是从1开始。

重点：句柄约等于唯一标识符，一旦OPC组或者OPC标签在OPC服务器里添加成功， OPC服务器将赋予被添加的各个OPC组或者各个OPC标签一个独特的标识符。 被赋予的服务器句柄将返回给OPC应用程序。

OPC应用程序应该将由OPC服务器返回的OPC组或者OPC标签的服务器句柄好好保管。 因为随后对添加的OPC组或者OPC标签进行操作时，只有使用这些服务器句柄才可以唯一地识别特定的OPC组或者OPC标签。

3.9周三

里氏转换：子类（信息多）可以赋值给父类 ，父类装子类对象，可以强转为子类对象。person是父类，student是子类。可以Person p = new Student（）。

ArrayList和HashTable

ArrayList：相比数组，数组定义完个数不能变，且arraylist添加类型多样。count属性表示实际存在的，capacity表示容量。可变！！4->8->16->32不断申请一倍的空间。（C++里的容器）,拿出来都是object类型，需要转换。

HashTable：键值对（python中的字典），键值类型任意，但键必须唯一。ht.Add(key,value) 遍历：foreach(var item in ht.Keys) 输出ht[item]。

File类读写：ReadAlllines（一行行读），ReadAllbytes,WriteAlllines。Filestream类，是操作字节的。StreamReader是操作字符的。打开文件流，必须关闭再释放，fs.Close()，fs.dispose()。\*将创建文件流对象的过程写在using中，会自动帮我们释放流所占的资源。

例：using(filestream fs = new filestream(参数1，参数2，参数3，参数4){读方法or写方法}

泛型：List<type> list =new List<type>() 集合转数组 toArray ，数组转集合toList

Dictionary类（对应hashtable），确定的键类型和值类型

装箱：值->引用 拆箱：引用->值。是否发生装、拆箱要看两种类型是否是继承关系。这种慢，尽量避免。（c#P120-P150）

3.10周四

多态：让一个对象能够表现出多种状态。实现多态的三种方法：1、虚方法（父类方法加virtual，子类加override） 2、抽象类（abstract，抽象类没有方法体，且不能实例化） 3、接口

访问修饰符：能修饰类的只有public 和 internal（默认）

internal：只能在当前项目访问。子类的访问权限不能高于父类的访问权限。

序列化：将对象转换为二进制 反序列化：二进制转为对象 作用：传输数据 关键字，在类上 [Serializable]，序列化对象BinaryFormatter

partial部分类，一个项目多个人共写一个类，partial加起来就是一个完整的类。

董老师给了资料关于C# 界面UI优化的库：NuGet程序包中找MetroModernUI。可以用控件做出美化的C#界面

3.11周五

上午：学习C# winform窗体

1.容器概念：panel一般嵌套窗体，groupbox放组件。容器有controls属性（用于多选和遍历同一类的物）

2.DateTimePicker事件获取器件，默认是中文，必须转到标准格式（CustomerFormat属性中设置（yyyy-MM-dd HH:mm:ss) ) 3.MaskedTextbox(多用于掩藏密码）

4.numericalUpDown——小范围输入且限定为数字

5.PictureBox 图片控件，image可以用作gif

6.radiobutton单选按钮（类比checkedbox）互斥，通过groupbox可以做同一容器内互斥

7.ComboBox下拉列表，如果想设置为只能选下拉的，属性DropdownStyle设置为list。有动态添加（数据库读）和固定（自定义list）两种方式

8.tabcontrol容器，多用于同一类的分块。

9.菜单栏：menustrip——就跟左上角文件-编辑-格式 一样。右键设置图像可以设图在文字前

10\*数据——DataGridView（数据显示控件用于动态删除，更新数据-按行） 其数据源是集合对象（list，datasource）//禁止自动生成列.AutoGenerateColumns=false;

----------------------------------------------------------------------------------------------------------------------------------------

进阶：自主开发封装控件

C#必备控件库：添加-用户控件（窗体）

搞清卷号、捆包号、尾卷号、标定卷的关系，与数据库哪张表有联系。

3.12周六

密封类（sealed）可以继承别人，但不能被其他类继承

接口语法：public interface 名称{方法}，接口中成员不允许添加访问修饰符，默认为public，且没有方法体，不允许有字段。关键字：能力！

3.13周日

winform：

textbox控件——worwrap：指示是否换行 scollbars：显示滚动条

timer——定时控件，间隔触发

Mdi窗体设计：子窗体全在一个父窗体内，且拖拽不离父窗体 IsMainContainer = true（置为父窗体），menustrip——菜单，设置 子窗体.Mdiparent = this

3.14周一

上午：看spm画面程序，子卷号和卷号的关系。钢卷跟踪这一界面的构造：包括逻辑显示，看了一半，按键部分还有一部分。

3.15周二

钢卷跟踪这一块总算看完了，对各电文表达含义以及变量涉及到数据库表的变化有了更为深刻的理解。重看一遍过程自动化跟踪设计的文档，明朗了很多。

3.16周三

一早遛狗发现小区被封了。在家办公，打算趁这机会把C#啃完，C++开始入门了。

C#GDI+绘图：是一种绘图装置接口 关键字 graphics 类似python的turtle

可以弄验证码 （位图bitmap、graphics、随机数）

html+css复习

html：p是段落标签，上下有空隙<p></p> <br/> 换行（无空隙）

a标签：

①超链接 <a href = "www.baidu.com" >百度</a>

②页面跳转 <a name="顶端" href=#"底端">回到底端</a> <a name="底端" href=#"顶端">回到顶端</a>

③发邮件 <a href="mailto:865944203@qq.com">给lxw发邮件</a>

CSS：div+span（网页布局）大的div嵌套小的div

3.17周四

C#复习：抽象类 其子类必须继承它所有的抽象方法

父类是单继承，接口可以多继承

设计模式：简单工厂（返回一个父类对象）

c++视频：p10-20

3.18到3.20

把spm画面程序整整齐齐看了一遍，归纳总结

下午看董老师推荐的C# ui设计——MetroModernUI。

图标可以从阿里巴巴矢量图标库获取！！！挺好用的感觉

dock属性可以设置停靠位置

多画面切换代码：void OpenForm(Form frm)

{

frm.TopLevel=false;

frm.TopMost=false:

this.panelMain.Control.Clear();//清除当前控件，防止叠加

this.panelMain.Control.Add(frm);

frm.Show();

}

panel就是一个块，可以当容器，panel也可以用来设置滚动条

序列化是通过将对象转换为字节流，从而存储对象或将对象传输到内存，数据库或文件的过程。主要用途是保存对象的状态，包括对象的数据，以便能够在需要是重建对象。反向过程称为 反序列化。

序列化的用途：

通过序列化，可以执行如下操作：通过 Web 服务将对象发送到远程应用程序、在域之间传递对象、以 XML 字符串的形式传递对象通过防火墙、跨应用程序维护安全性或用户专属信息。

03.21

上午：继续学习C++

下午：UI设计继续，工具箱里chart可以根据plc获取的数据画出实时图形。数据是由series表示的，legend修改图例。

03.22

上午：C#复习，若进行大量字符串拼接操作需使用stringbuilder，但stringbuilder仅仅是拼接字符串的工具，大多数情况下最后还是要把stringbuilder转成string

集合复习：四种 和 拆装箱 装箱：值转引用 拆箱：引用转值 拆装箱的两种类型必须具有继承关系

List、Arraylist（类数组） Dictionary、HashTable （键值对）

静态类static构造函数只运行一次

结构和类都可以有构造函数，但结构的构造函数必须包含结构中所有参数（所以不能重载），结构new还是在栈中开辟空间（值类型）

File复习：file会读取所有，一行一行读，数组存储。filestream负荷小（适用于读大数据），要dispose，常与using搭配使用 using（FileStream fs = new Fi了stream（@“文件名”,mode,access)) 读和写都要使用字节数组

序列化：在需要序列化的类前加上[serializable]，要和BinaryFormatter搭配使用

winform中：如果要设置右键菜单栏，可以用contextMenuStrip控件，记得要和使用的绑定起来。

03.24

上午：学习C#中进阶概念——泛型和反射

下午：修改代码逻辑SPM，整理文档

晚上：C#学习视频完结（xml和委托部分）

html用于显示，xml用于存储信息（轻型数据库）using System.Xml;xml中必须有且仅有一个根元素

03.25

C# mvvm和WPF学习

03.26

指针学习，今天自己捅了自己做了抗原检测

03.27

这周又是家中办公，继续学习。上海浦东、浦西分批次封闭管理。浦西4.2开始，商铺物价爆涨，附近人开始扫荡了。

03.28

C++学习 ，征服C指针书看完

03.29

发现了宝藏C#视频，刘铁猛老师的，确实进阶，确实精简

打算再滚一遍C#。顺便WPF也看他了。

03.30-03.31

向董老师汇报总结成果，并改进

刘铁猛WPF基础视频完结